IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 9, NO. 6, SEPTEMBER 1999

Bit Allocation and Constraints for Joint Coding of Multiple Video Programs

PD. 00/09/99
p.949-959 = (11)

Limin Wang and André Vincent

VN. 8378 XP-000848859

Abstract—Recent studies have shown that joint coding is more efficient and effective than independent coding for compression of multiple video programs [3]–[7]. Unlike independent coding, joint coding is able to dynamically distribute the channel capacity among video programs according to their respective complexities and hence achieve a more uniform picture quality. This paper examines the bit-allocation issues for joint coding of multiple video programs and provides a bit-allocation strategy that results in a uniform picture quality among programs as well as within a program. To prevent the encoder/decoder buffers from overflowing and underflowing, further constraints on bit allocation are also discussed.

Index Terms—Constraints for buffer protection, dynamic bit allocation, joint coding, multiprogram transmission, rate control, statistical multiplexing.

I. Introduction

With recent advances in digital video compression, such as MPEG-2 [1], and in digital transmission technology, conventional television channels are now capable of delivering several digitally compressed video programs. For example, by using MPEG-2 video coding, digital NTSC video programs can be compressed to 4–8 Mbits/s while providing an adequate picture quality. Digital transmission technology can support a payload of up to 19 Mbits/s in a 6-MHz terrestrial broadcast TV channel or 27–43 Mbits/s on cable TV channels. Hence, it is possible to transmit several digital television programs in the same bandwidth presently occupied by a single analog TV channel.

Fig. 1 shows a block diagram of a multiprogram transmission system in which several video programs are compressed, multiplexed, and transmitted over a single channel. Clearly, these programs have to share the channel capacity. In other words, the aggregate bit rate of the programs has to be equal to (or less than) the channel rate. This can be achieved by controlling either each individual program bit rate (independent coding) or the aggregate bit rate (joint coding).

In independent coding, programs are coded independently. Each encoder, therefore, needs a separate rate control to maintain its bit stream at a constant bit rate (CBR), as shown in Fig. 1. Rate control for independent coding can be performed across the time and spatial dimensions of a program. There are two main shortcomings with independent coding at a constant bit rate. First, independent coding at a fixed rate may lead to large variations in picture quality among programs. Program complexity typically varies significantly between programs. For example, news programs usually have a relatively static background and very little motion, while sport and music video programs may contain fast motions and rapid scene changes. As an example, Fig. 2 shows the peak signal-to-noise ratio (PSNR) for three video programs (*Ayers*, *Martin*, and *Hook*) coded independently using MPEG-2 at a bit rate of 3 Mbits/s. It is clearly demonstrated that the picture quality level among these three programs is quite different. The second problem with independent coding at a constant rate is an inefficient use of the channel capacity. Not all scenes within a program need to be encoded at the same bit rate to achieve a similar picture quality. For example, relatively static scenes require much less bits than complex scenes with fast motion. If the bit rate for a given program is set sufficiently high to maintain a good picture quality for the most demanding segment, that bit rate is effectively wasted during the less challenging periods. Furthermore, if a fixed bit rate is used, different scenes will be rendered with unequal quality, as shown in Fig. 2, where the PSNR for each program also varies with time.

Unlike independent coding, joint coding only needs to maintain the aggregate bit-rate constant, while allowing the bit rate of each individual program to vary, as shown in Fig. 3. Rate control for joint coding is therefore extended to an additional dimension, that is, the program dimension, which implies more freedom in allocating the channel capacity among programs and hence more control over the picture quality among programs, as well as within a program. For example, joint coding can move bits from the less active programs to the more active ones in order to maintain a uniform picture quality. Recent studies have shown that joint coding is more efficient and effective than independent coding for multiprogram video coding [3]–[7]. In fact, the study in [5] shows that in joint coding, as the number of programs increases, the compressed bit streams exhibit characteristics similar to those of variable bit-rate compression, which is known to be more efficient than CBR compression [3], [7].

This paper examines the bit-allocation issue for joint coding of multiple video programs. In Section II, we present a bit-allocation strategy for uniform picture quality. The bit-allocation strategy is of a hierarchical nature. At the very top level, we introduce the concept of a super group of pictures

Manuscript received January 8, 1998; revised March 12, 1999. This paper was recommended by Associate Editor J.-N. Hwang.

L. Wang is with General Instrument Corp., San Diego, CA 92121 USA (e-mail: lwang@gi.com).

A. Vincent is with the Communications Research Centre, Ottawa, Ont. K2H 8S2 Canada (e-mail: andre.vincent@crc.doc.ca).

Publisher Item Identifier S 1051-8215(99)07025-1.

1051-8215/99$10.00 © 1999 IEEE

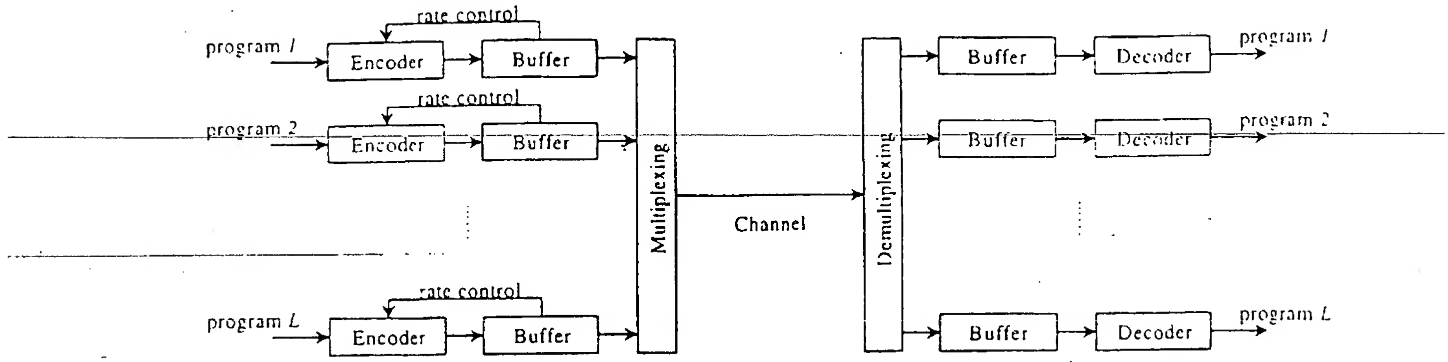

Fig. 1. Independent coding: programs are coded separately, each with a separate rate control.

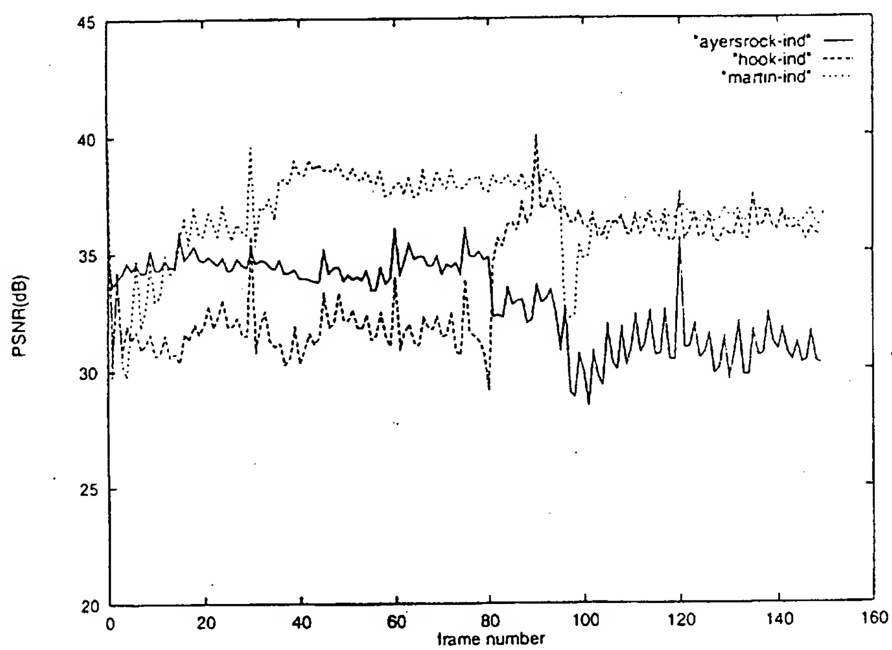

Fig. 2. The PSNR for three sequences (*Ayers*, *Martin*, and *Hook*) using independent coding.

(GOP) in Section II-A. The input video programs are divided into super GOP's that are identical in terms of the number of frames of each type (I, P, and B). The super GOP's are all assigned the same number of bits. In Section II-B, we define a super frame as a collection of frames, one from each of the programs taken at the same time instant. Given a number of bits for a super GOP, we formulate the bit-allocation strategy for each super frame within the super GOP. To prevent the encoder buffer from overflowing and underflowing, we impose constraints on the target number of bits for each super frame, as discussed in Section II-C. In Section II-D, the same methodology is extended to the frame level, where each regular frame receives a target number of bits that is proportional to its complexity measure. To ensure that the decoder buffer never overflows or underflows and to limit the bit rate of each individual program within a specific range, we apply additional constraints on the target number of bits for each regular frame. In Section III, we report the simulation results.

II. Dynamic Bit Allocation

Assume that all the encoder engines in joint coding (Fig. 3) use MPEG-2 video coding [1]. The MPEG-2 standard defines three main picture types in terms of temporal processing: I (intraframe coded) pictures, P (forward predictive coded) pictures, and B (bidirectional predictive coded) pictures. The input video frames are organized into GOP's; each may contain one I-picture, a number of P-pictures, and optionally some B-pictures between I- or P-pictures. Fig. 4 shows an example where the input video sequence is divided into GOP's of 15 frames: one I-picture, four P-pictures, and two B-pictures between I- or P-pictures.

A. Super GOP and Target Rate

Assume that L video programs are to be delivered over a network with a fixed channel rate $R_{channel}$, as shown in Fig. 3. The L programs can have any GOP structure, that is,

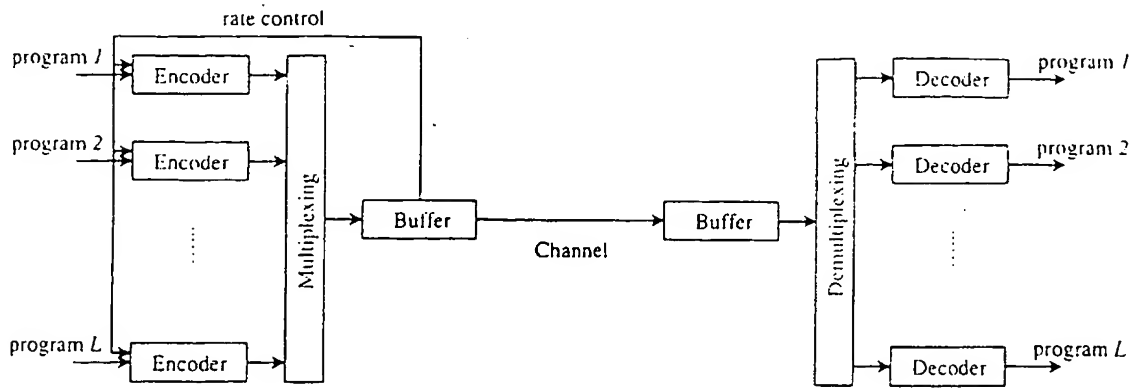

Fig. 3. Joint coding: programs are coded jointly; a joint rate control is implemented.

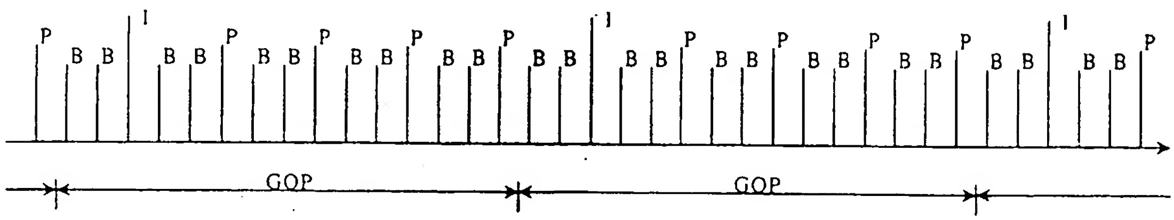

Fig. 4. The input video sequence is divided into GOP's.

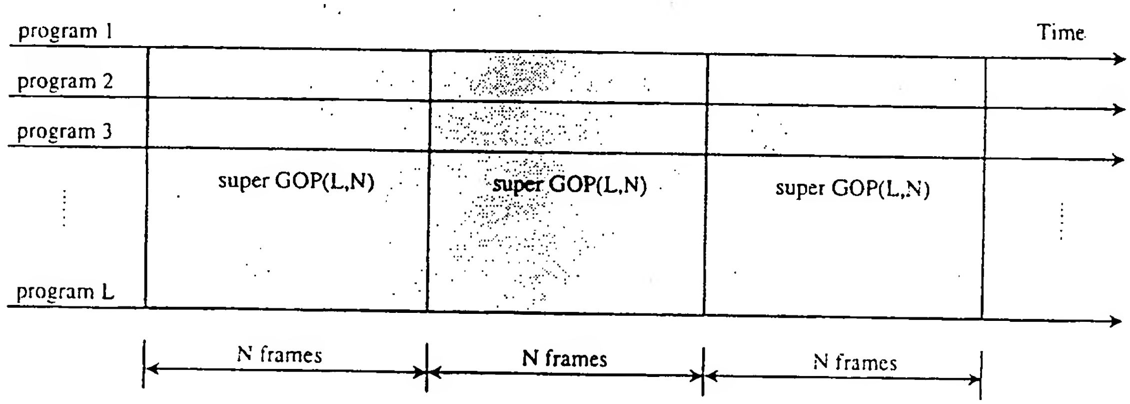

Fig. 5. The L programs are conceptually divided into smallest identical groups, so-called super GOP's(L, N). Each contains $L \times N$ frames, where L is the number of programs and N is the least common multiple of program GOP lengths.

the program GOP length as well as the distance between I- or P-pictures can be different from program to program.

We start by conceptually dividing the L programs into identical groups, so-called super GOP(L, N), as shown in Fig. 5, in terms of the number of frames of each picture type. Here, L is the number of programs and N is the length of super GOP's. Since these super GOP's contain the same number of frames of each picture type, they are assigned the same number of bits. Clearly, each super GOP(L, N) contains $L \times N$ frames. As a matter of fact, the value of N is not unique, but from an implementation point of view, small super GOP's are preferable. Let N_l, $l = 1, 2, \ldots, L$ be the GOP length for program l. We set N equal to the least common multiple (LCM) of N_l, $l = 1, 2, \ldots, L$, i.e.,

$$N = \mathrm{LCM}(N_1, N_2, \ldots, N_L) \tag{1}$$

where N is the smallest number that can be divided by all N_l, $l = 1, 2, \ldots, L$. Hence, the super GOP's(L, N) are the smallest identical groups containing the same number of frames of each picture type. For example, if there are two different program GOP lengths for these L programs, say 9 and 15, we will have a super GOP length of 45 frames.

A super GOP always contains an integer number of GOP's of a program. The GOP's of individual programs, however, do not have to be aligned with each other or with the super GOP. A super GOP boundary may cut through a program GOP, or a super GOP may include fractional portions of the GOP's of a program at the boundaries. The fractional portions of GOP's of a program included inside a super GOP, at the super GOP's left and right boundaries, can however be considered as a "complete" GOP. Fig. 6 shows an example of a super GOP

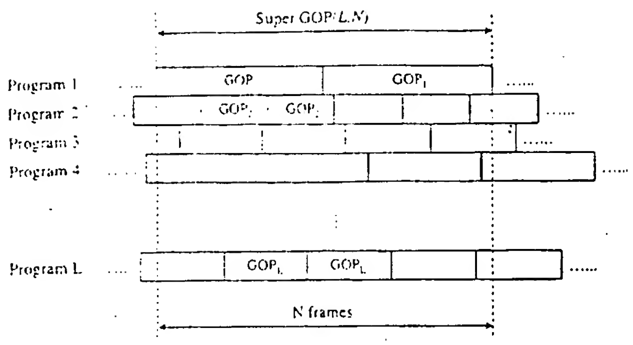

Fig. 6. A super GOP may include fractional portions of GOP's of a program at boundaries. The fractional portions of GOP's of a program included inside a super GOP, at super GOP boundaries, can be considered as a "complete" GOP. A super GOP contains an integer number of GOP's of each program.

with nonaligned program GOP's. As seen, the boundaries of the GOP's of program 2 do not coincide with the boundaries of the super GOP. However, the two fractional parts of the GOP's included inside the super GOP, at the super GOP's left and right boundaries, actually contain the same numbers of I-, P-, and B-pictures as a complete GOP. Furthermore, if all the programs have the same GOP length, say, N, the super GOP length will also be equal to N, regardless of whether the program GOP's have the same patterns of I-, P-, and B-pictures or whether they are synchronized or not. In any case, super GOP's contain the same number of frames of each picture type from one of programs.

Since all super GOP's(L, N), with N defined by using (1) contain the same number of I-, P-, and B-pictures, they can be assigned the same number of bits T, i.e.,

$$T = N R_{\text{channel(bpf)}} \tag{2}$$

where $R_{\text{channel(bpf)}} = \mathit{channel\_rate}/\mathit{frame\_rate}$ is the average number of bits per frame.

B. Super Frames and Target Rate

Given the target number of bits T, for a super GOP(L, N), the next step is to determine the distribution of T over the frames within a super GOP. We further define a super frame as a collection of L frames, one from each of the L programs taken at the same time instant, as shown in Fig. 7. Clearly, a super GOP contains N super frames. Note that these L programs can have any GOP structure, and the L frames in a super frame may be of any picture type. In the following, we will determine a target number of bits for a super frame and then a target number of bits for a regular frame according to the program complexities.

As in MPEG-2 Test Model 5 (TM5) [2], we define a complexity measure C for a frame as the product of the quantization Q used for the frame and the number of bits R generated for the frame using Q, i.e.,

$$C = RQ. \tag{3}$$

Let $Q_{l,n,t}$ and $R_{l,n,t}$ be, respectively, the quantization parameter used for frame n of program l and the corresponding number of bits generated for the frame using $Q_{l,n,t}$, where t indicates the picture type I, P, and B. For a super frame n, there can be L different frame complexity measures, one for each regular frame, i.e.,

$$C_{l,n,t} = Q_{l,n,t} R_{l,n,t} \qquad l = 1, 2, \ldots, L. \tag{4}$$

Let T_n be the target number of bits for super frame n. The total number of bits generated from the L regular frames within super frame n should be equal (or close) to T_n, i.e.,

$$T_n = \sum_{l=1}^{L} R_{l,n,t} = \sum_{l=1}^{L} \frac{C_{l,n,t}}{Q_{l,n,t}}. \tag{5}$$

Similarly, the total number of bits generated for all the super frames in a super GOP should be equal (or close) to the target number of bits T assigned for each super GOP, i.e.,

$$T = \sum_{n=1}^{N} T_n. \tag{6}$$

Since one of the objectives of joint coding is to achieve a uniform picture quality, ideally, the same quantization parameter should be applied to all the frames (note that quantization is the only lossy operation in MPEG-2 encoding and plays a critical role in controlling both the picture quality and the bit rate). However, in order to take into consideration the different picture types (I, P, and B), as in TM5, we introduce a constant weighting factor $K_{l,n,t}$ for each picture type t, i.e.,

$$Q_{l,n,t} = K_{l,n,t} Q \tag{7}$$

where

$$K_{l,n,t} = \begin{cases} K_{\mathrm{I}} & \text{for I picture} \\ K_{\mathrm{P}} & \text{for P picture} \\ K_{\mathrm{B}} & \text{for B picture.} \end{cases} \tag{8}$$

Note that t corresponds to the picture type, I, P, or B. In TM5, $K_{\mathrm{I}} = K_{\mathrm{P}} = 1$ and $K_{\mathrm{B}} = 1.4$. From (5) to (7), we can obtain the target number of bits for super frame n, T_n as

$$T_n = \frac{\sum_{l=1}^{L} \frac{1}{K_{l,n,t}} C_{l,n,t}}{\sum_{n=1}^{N} \sum_{l=1}^{L} \frac{1}{K_{l,n,t}} C_{l,n,t}} T \tag{9}$$

where the numerator is the sum of the complexity measures for the L regular frames in super frame n and the denominator is the sum of the complexity measures for all the frames in the current super GOP. Clearly, the target number of bits for super frame n, T_n, is proportional to its complexity.

The problem with the above bit-allocation equation is that in computing a target number of bits for a super frame, we need to have the complexity measures for all the $L \times N$ frames within the current super GOP(L, N), i.e., $C_{l,n,t}$, $l = 1, 2, \ldots, L$ and $n = 1, 2, \ldots, N$, which may not be very practical. We now make two modifications to simplify the calculation. First, at

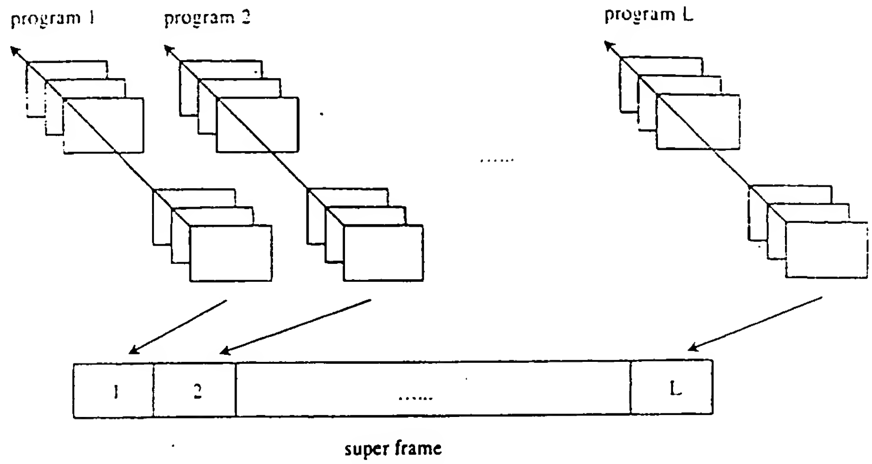

Fig. 7. A super frame is a collection of L frames, one from each of L programs at the same time instant.

each super frame n, we only consider the distribution of the remaining bits T_r defined as

$$T_r = T_r - \sum_{l=1}^{L} R_{l,n-1,t} \tag{10}$$

over the remaining super frames from n to N in the super GOP(L, N). Here, $\sum_{l=1}^{L} R_{l,n-1,t}$ is the number of bits generated from super frame $n-1$. This leads to the bit allocation for super frame n as

$$T_n = \frac{\sum_{l=1}^{L} \frac{1}{K_{l,n,t}} C_{l,n,t}}{\sum_{n=n'}^{N} \sum_{l=1}^{L} \frac{1}{K_{l,n,t}} C_{l,n,t}} T_r. \tag{11}$$

The complexity measures for the previous frames from 1 to $n-1$ are now no longer necessary in computing the target number of bits for super frame n, T_n. At the start of processing a new super GOP, T_r needs to be reset as follows:

$$T_r = T_r + T \tag{12}$$

where T is the target number of bits for a new super GOP (2) and T_r on the right of the equation is actually the number of bits left over from the previous super GOP, which can be either a positive or negative number.

Secondly, we assume that all the future frames of the same picture type in a program have the same complexity measure, i.e.,

$$C_{l,n',t} = C_{l,n,t} \qquad n \le n' \le N \tag{13}$$

which is a reasonable assumption for continuous scenes. Now, for each program, we only need to keep three complexity measures, one for each of the three picture types I, P, or B, i.e., $C_{l,\mathrm{I}}$, $C_{l,\mathrm{P}}$, and $C_{l,\mathrm{B}}$. The bit-allocation strategy for super frame n therefore becomes

$$T_n = \frac{\sum_{l=1}^{L} \frac{1}{K_{l,n,t}} C_{l,n,t}}{\sum_{l=1}^{L} \left[N_{l,\mathrm{I}} \frac{C_{l,\mathrm{I}}}{K_{\mathrm{I}}} + N_{l,\mathrm{P}} \frac{C_{l,\mathrm{P}}}{K_{\mathrm{P}}} + N_{l,\mathrm{B}} \frac{C_{l,\mathrm{B}}}{K_{\mathrm{B}}} \right]} T_r \tag{14}$$

where:

1) $C_{l,n,t}$ is the complexity measure for frame n of picture type $t \in \{\mathrm{I, P, B}\}$ for program l. After encoding frame n, the complexity measure for the corresponding picture type $t \in \{\mathrm{I, P, B}\}$ of program l, $C_{l,t \in \{\mathrm{I,P,B}\}}$ is updated. It can be estimated/calculated based upon the average quantization parameters used for the frame and the number of bits generated for the frame (3). The updated complexity measures are then used in calculating the target rates for the next frames.
2) $K_{l,n,t}$ is a constant factor used to compensate for the picture type $t \in \{\mathrm{I, P, B}\}$ of frame n of program l. It can be either K_{I}, K_{P} or K_{B}, depending upon the picture type.
3) $N_{l,\mathrm{I}}$, $N_{l,\mathrm{P}}$, and $N_{l,\mathrm{B}}$ are, respectively, the remaining numbers of I, P, and B pictures for program l in the current super GOP at super frame n.

The numerator on the right side of (14) is the sum of complexity measures for all the frames in super frame n. It can be considered as a complexity measure for super frame n. The denominator can, on the other hand, be considered as a complexity measure for the entire set of the remaining frames in the super GOP. Hence, we can state that (14) actually assigns a super frame a target number of bits proportional to the super frame's complexity measure.

C. Constraint on Super Frame Target Rate

For transmission over a constant bit-rate channel, the aggregate bit stream needs to be smoothed out using the encoder channel buffer, as shown in Fig. 3. The encoder buffer fullness

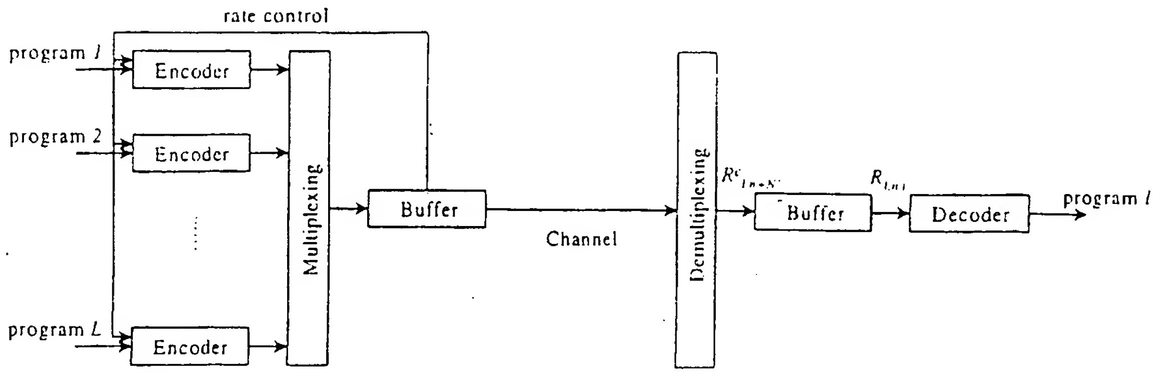

Fig. 8. Packets of selected program are extracted and decoded at the decoding end.

at frame n, B_n^e (Fig. 3) can be calculated as [8]

$$B_n^e = B_{n-1}^e + \sum_{l=1}^{L} R_{l,n,t} - R_{\text{channel(bpf)}}. \tag{15}$$

Let $B_{\max}^e$ be the maximum encoder buffer size. To ensure that the encoder buffer never overflows or underflows, the buffer fullness B_n^e has to be constrained within the range $[0, B_{\max}^e]$, i.e.,

$$0 \le B_n^e \le B_{\max}^e. \tag{16}$$

From (15) and (16), we have

$$0 \le B_{n-1}^e + \sum_{l=1}^{L} R_{l,n,t} - R_{\text{channel(bpf)}} \le B_{\max}^e \tag{17}$$

or

$$R_{\text{channel(bpf)}} - B_{n-1}^e \le \sum_{l=1}^{L} R_{l,n,t} \le R_{\text{channel(bpf)}} + B_{\max}^e - B_{n-1}^e. \tag{18}$$

This is a constraint on the total number of bits generated for super frame n for a given channel rate $R_{\text{channel(bpf)}}$. If the aggregate rate can be controlled to meet the target rate, i.e.,

$$T_n = \sum_{l=1}^{L} R_{l,n,t} \tag{19}$$

the constraint on the total number of bits for a super frame n becomes the constraint on the target number of bits for super frame, i.e.,

$$R_{\text{channel(bpf)}} - B_{n-1}^e \le T_n \le R_{\text{channel(bpf)}} + B_{\max}^e - B_{n-1}^e. \tag{20}$$

Hence, before starting to encode each super frame n, we need to check if its target rate determined by (14) is in the proper range, and if not, we adjust it as follows:

$$T_n = \begin{cases} R_{\text{channel(bpf)}} - B_{n-1}^e, & \text{if } T_n < R_{\text{channel(bpf)}} - B_{n-1}^e \\ R_{\text{channel(bpf)}} + B_{\max}^e - B_{n-1}^e; & \text{if } T_n > R_{\text{channel(bpf)}} + B_{\max}^e - B_{n-1}^e \\ T_n; & \text{otherwise.} \end{cases} \tag{21}$$

D. Target Rate for Regular Frames

Once a target number of bits for a super frame n, T_n, has been set, what remains is to distribute the bits over the regular frames in the super frame. By the same methodology, we can have the target number of bits for frame n of program l, $T_{l,n}$, as

$$T_{l,n} = \frac{\dfrac{1}{K_{l,n,t}} C_{l,n,t}}{\displaystyle\sum_{l=1}^{L} \frac{1}{K_{l,n,t}} C_{l,n,t}} T_n \tag{22}$$

where the numerator on the right is the complexity measure for frame n of program l and the denominator is the complexity measure for super frame n. The distribution of T_n assigned for super frame n over L regular frames in the super frame is again based upon the complexity measures of the program frames.

E. Constraint on Target Rate for Regular Frame

In multiprogram broadcast applications, several video programs are multiplexed on a single fixed-rate transmission channel. Service information included in the bitstream would provide the necessary navigation information to allow the viewer to select the desired programs and the settop to turn onto the proper channel and extract (demultiplex) the packets corresponding to the selected programs, as shown in Fig. 8. The demultiplexed bitstream is at a variable rate. To ensure that the decoder buffer does not overflow or underflow when any of the programs are selected, we need to apply, at the encoder, additional restrictions to each individual program's bit rate.

Assume that program l is selected and that the decoding delay is N' frames. Let $R_{l,n}^c$ be the number of bits transmitted for program l during the nth frame period. The decoder buffer will therefore be filled up to

$$B_0^d = \sum_{n'=1}^{N'} R_{l,n}^c \tag{23}$$

before any bits are moved out. During the period of frame $n + N' > N'$, the decoder buffer moves $R_{l,n,t}$ bits to the

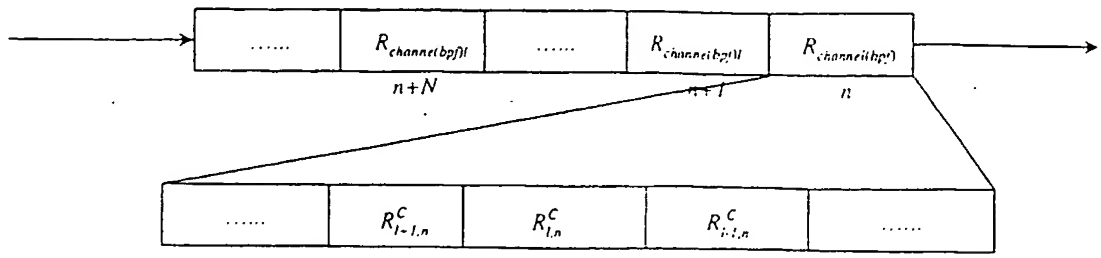

Fig. 9. Encoder buffer: within each slot window $n' = n, n+1, \ldots, n+N'$ of $R_{\text{channel(bpf)}}$ bits, the number of bits for program l is $R^c_{l,n'}$.

decoder and receives $R^c_{l,n-N'}$ bits from the network, as shown in Fig. 8. The buffer fullness at frame $n+N' > N'$ is therefore given as

$$B^d_n = B^d_{n-1} + R^c_{l,n+N'} - R_{l,n,t} \qquad n = 1, 2, \ldots \tag{24a}$$

$$= B^d_0 + \sum_{n'=1}^{n} R^c_{l,n'+N'} - \sum_{n'=1}^{n} R_{l,n',t} \tag{24b}$$

$$= \sum_{n'=1}^{N'} R^c_{l,n'} + \sum_{n'=1+N'}^{n+N'} R^c_{l,n'} - \sum_{n'=1}^{n} R_{l,n',t} \tag{24c}$$

$$= \sum_{n'=n}^{n+N'} R^c_{l,n'} + \left(\sum_{n'=1}^{n-1} R^c_{n'=1} R_{l,n',t} \right) - R_{l,n,t} \tag{24d}$$

$$= \sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1} - R_{l,n,t} \tag{24e}$$

where

$$B^e_{l,n} = \sum_{n'=1}^{n} R_{l,n',t} - \sum_{n'=1}^{n} R^c_{l,n} \tag{25}$$

can be considered as the fullness of a virtual encoder buffer for program l at frame n. Let $B^d_{\max}$ be the maximum decoder buffer size. To ensure that the decoder buffer never overflows or underflows, the buffer fullness B^d_n has to be within the range of $[0, B^d_{\max}]$, i.e.,

$$0 \le B^d_n \le B^d_{\max}. \tag{26}$$

From (24) and (26), we have

$$0 \le \sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1} - R_{l,n,t} \le B^d_{\max} \tag{27}$$

or

$$\sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1} - B^d_{\max} \le R_{l,n,t} \le \sum_{n'=n}^{n+N'} R^c_{l,n} - B^e_{l,n-1}. \tag{28}$$

This is a constraint on the number of bits generated for frame n of program l. Again, we assume that the bit rate for each program can be controlled to meet its target rate, i.e.,

$$T_{l,n} = R_{l,n,t}. \tag{29}$$

The constraint on the bit rate for each individual frame (l, n) becomes the constraint on the target number of bits for the frame, i.e.,

$$\sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1} - B^d_{\max} \le T_{l,n} \le \sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1}. \tag{30}$$

Hence, before starting to encode each frame n of program l, we need to check if its target rate is within the proper range, and if not, we adjust it as follows:

$$T_{l,n} = \begin{cases} \sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1} - B^d_{\max}, & \text{if } T_{l,n} < \sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1} - B^d_{\max} \\ \sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1}, & \text{if } T_{l,n} > \sum_{n'=n}^{n+N'} R^c_{l,n'} - B^e_{l,n-1} \\ T_{l,n}, & \text{otherwise.} \end{cases} \tag{31}$$

Here, $B^e_{l,n-1}$ is the fullness of a virtual encoder buffer for program l at frame $n-1$ and is available at frame n. However, $R^c_{l,n'}, n' = n, n+1, \ldots, n+N'$, are the number of bits that will be transmitted for program l during the intervals of the current and future frames $n, n+1, \ldots, n+N'$. Fortunately, in our case, the channel rate $R_{\text{channel(bpf)}}$ is constant in bits/frame. So, we are able to measure these $R^c_{l,n'}, n' = n, n+1, \ldots, n+N'$, in the encoder buffer, as shown in Fig. 9. Within each slot window $n = n, n+1, \ldots, n+N'$ of $R_{\text{channel(bpf)}}$ bits in the encoder buffer, the number of bits for program l is $R^c_{l,n}$. Note that $R^c_{l,n}$ is not necessary nonzero.

F. Constraint on Max and Min Rate

We can also control the average bit rate over a certain number of frames by limiting the target number of bits for each frame within a specific range. Let $R_{\max}$ and $R_{\min}$ be the allowed maximum and minimum average bit rates over N'' frames. The average bit rate of N'' frames up to a frame n therefore has to be in the range of $[R_{\min}, R_{\max}]$, i.e.,

$$R_{\min} \le \frac{1}{N''} \sum_{n'=n-N''-1}^{n} R_{l,n',t} \le R_{\max} \tag{32}$$

TABLE I
GOP STRUCTURES USED FOR TEST SEQUENCES

GOP	*Ayers*	*Hook*	*Martin*	*Flower*	*Mobile*	*Tennis*
Length	15	15	18	15	18	15
# of B	2	2	2	2	2	2

TABLE II
PROGRAMS INCLUDED AT EACH STAGE

L	*Ayers*	*Hook*	*Martin*	*Flower*	*Mobile*	*Tennis*
3	Yes	Yes	Yes			
4	Yes	Yes	Yes	Yes		
6	Yes	Yes	Yes	Yes	Yes	Yes

or

$$N''R_{\min} - \sum_{n'=n-N''+1}^{n-1} R_{l,n',t} \le R_{l,n,t} \le N''R_{\max} - \sum_{n'=n-N''+1}^{n-1} R_{l,n',t}. \quad (33)$$

This is an additional constraint on the number of bits for frame n of program l. Note that $R_{l,n',t}$, $n' = n-N'', n-N'', \ldots, n-1$, are all available at frame n. Again, we assume that we can make the actual bit rate close to the target rate by proper rate control, i.e.,

$$T_{l,n} = R_{l,n,t}. \quad (34)$$

The additional constraint on the actual bit rate for frame n then becomes the constraint on its target rate, i.e.,

$$N''R_{\min} - \sum_{n'=n-N''+1}^{n-1} R_{l,n',t} \le T_{l,n} \le N''R_{\max} - \sum_{n'=n-N''+1}^{n-1} R_{l,n',t}. \quad (35)$$

If necessary, we adjust the target rate for each individual frame as follows:

$$T_{l,n} = \begin{cases} N''R_{\min} - \sum_{n'=n-N''_1}^{n-1} R_{l,n',t}, & \text{if } T_{l,n} < N''R_{\min} - \sum_{n'=n-N''+1}^{n-1} R_{l,n',t} \\ N''R_{\max} - \sum_{n'=n-N''-1}^{n-1} R_{l,n',t}, & \text{if } T_{l,n} > N''R_{\max} - \sum_{n'=n-N''_1}^{n-1} R_{l,n',t} \\ T_{l,n}, & \text{otherwise.} \end{cases} \quad (36)$$

III. SIMULATIONS AND RESULTS

In evaluating the bit-allocation strategy, simulations were carried out with a large number of test video programs. All the test materials had a spatial resolution of 720 × 480 with a color-sampling ratio of 4 : 2 : 2, interlaced at a frame rate of 30 Hz. The programs were encoded using the MPEG-2 main profile/main level syntax [1]. Table I shows the GOP structures used for the test sequences in the simulations. Some sequences have a GOP length of 18 frames, while the others have a GOP length of 15 frames. There are two B-pictures between I- or P-pictures.

For comparison purposes, the sequences were coded both independently and jointly. In independent coding, each sequence was encoded separately, at a bit rate of 3 Mbits/s, using the TM5 rate control [2]. In joint coding, the proposed bit-allocation strategy is used. To examine how the picture quality varies with respect to the number of programs, joint coding was performed with three different conditions—i.e., $L = 3, 4$, and 6 programs—at total bit rates of $3L$ Mbits/s, respectively. Sequences *Ayers*, *Hook*, and *Martin* were first grouped in a three-program joint coding at a total rate of 9 Mbits/s. The sequence *Flower* was then added to form a four-program joint coding, at a total rate of 12 Mbits/s. Two more sequences, *Mobile* and *Tennis*, were then included for a six-program joint coding. Table II summarizes the programs included at each step.

A. Simulations

Fig. 10 shows the block diagram of joint coding simulations. At each frame period, each MPEG-2 encoder, say, l, receives a target number of bits T_l from the rate-control engine. The target number of bits for each frame is then met by adjusting the quantization parameter in each MPEG-2 encoder. The resulting number of compressed bits (R_l) and the average quantization parameter (Q_l) used for the current frame of a program are sent to the rate-control engine, and their product is used to update the complexity measure for the corresponding picture type. The rate-control engine then determines a new target number of bits for the next frame, using the proposed bit-

TABLE III
BIT RATE IN Mbits/s

Coding	*Ayers*	*Hook*	*Martin*	*Flower*	*Mobile*	*Tennis*
Indep.	3.00	3.00	3.00	3.00	3.00	3.00
Joint 3	3.61	3.29	2.10			
Joint 4	3.24	2.98	1.88	3.89		
Joint 6	2.88	2.66	1.71	3.75	4.16	2.84

TABLE IV
PSNR (dB) FOR BOTH INDEPENDENT AND JOINT CODING

Coding	*Ayers*	*Hook*	*Martin*	*Flower*	*Mobile*	*Tennis*
Indep.	32.07	32.89	35.91	27.52	25.94	29.66
Joint 3	34.22	34.43	35.43			
Joint 4	33.56	33.87	34.90	31.43		
Joint 6	32.76	33.16	34.27	29.94	28.70	30.60

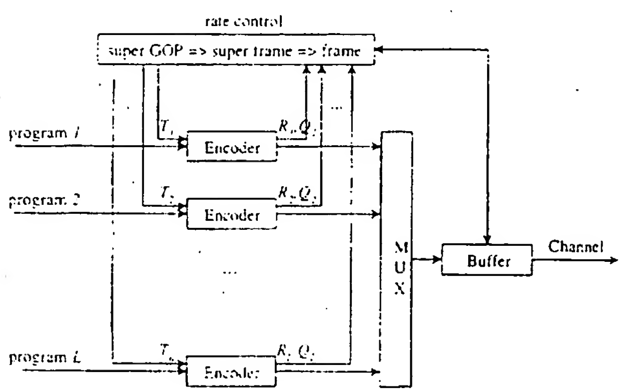

Fig. 10. Block diagram of joint coding with proposed joint rate.

allocation strategy described in Section II. The bits generated by the MPEG encoders are multiplexed and moved into the encoder buffer for transmission.

To achieve the target bit rate for each regular frame, rate control has to be implemented. TM5 [2] describes a rate-control scheme that adjusts the quantization parameter at each macroblock according to the buffer fullness. Since the quantization parameter varies from macroblock to macroblock, nonuniform picture quality may result. For example, two identical macroblocks in a frame may be assigned two different quantization parameters because of the different buffer fullness at different times. An alternative is to use a global quantization parameter for the entire frame. It has been shown [9] that the use of a unique quantization parameter over a frame results in a more uniform picture quality as compared to TM5. The global quantization parameter can be adjusted on a frame-by-frame basis so that the output bit rate for a frame can be close to the target bit rate for the frame. In our simulations, we used a binary tree search algorithm to determine the global quantization parameter for a frame [9]. To exploit spatial masking, the global quantization parameter can be further modulated for each macroblock by the respective local spatial activity (as in TM5 [2]). It has been shown [9] that the global rate control provides much better and more accurate rate control than TM5. The simulation results in [9] demonstrated that the deviation of the actual rate from the target rate by using the alternative is about 70–80% less than with TM5.

B. Results

Table III shows the bit rates for the test sequences for both independent and joint coding. In independent coding, all the sequences were coded at the same rate of 3 Mbits/s. However, in joint coding, the sequences were coded with quite different numbers of bits, depending on program contents. For example, with six programs, jointly coded, sequence *Mobile* used more than twice as many bits as sequence *Martin*.

Table IV shows the corresponding PSNR for each sequence. Compared with independent coding, joint coding produces a much smaller difference in PSNR between programs, even with a small number of programs. The variations in PSNR between programs depend on program contents.

Joint coding also produces a more uniform picture quality within programs. Table V shows the quality variation (variance of the PSNR) for the test sequences. It should be noted that even with a small number of programs, as few as three, joint coding significantly reduces time variations of picture quality, especially for programs that exhibit large variations in program content, such as *Ayers*, *Hook*, *Martin*, and *Tennis*. Simulations with more (eight and ten) programs have shown similar results, with only marginal improvements.

Figs. 11–13 show the comparisons in PSNR versus frame number for independent and joint coding with a different number of programs ($L = 3, 4, 6$). It is clearly demonstrated that the picture quality tends to be much closer with joint

TABLE V
VARIANCE IN PSNR FOR BOTH INDEPENDENT AND JOINT CODING

Coding	*Ayers*	*Hook*	*Martin*	*Flower*	*Mobile*	*Tennis*
Indep.	3.59	6.13	3.55	0.74	0.23	10.36
Joint 3	1.13	1.57	1.07			
Joint 4	0.82	1.31	0.88	0.80		
Joint 6	0.84	1.53	0.71	0.79	0.66	4.90

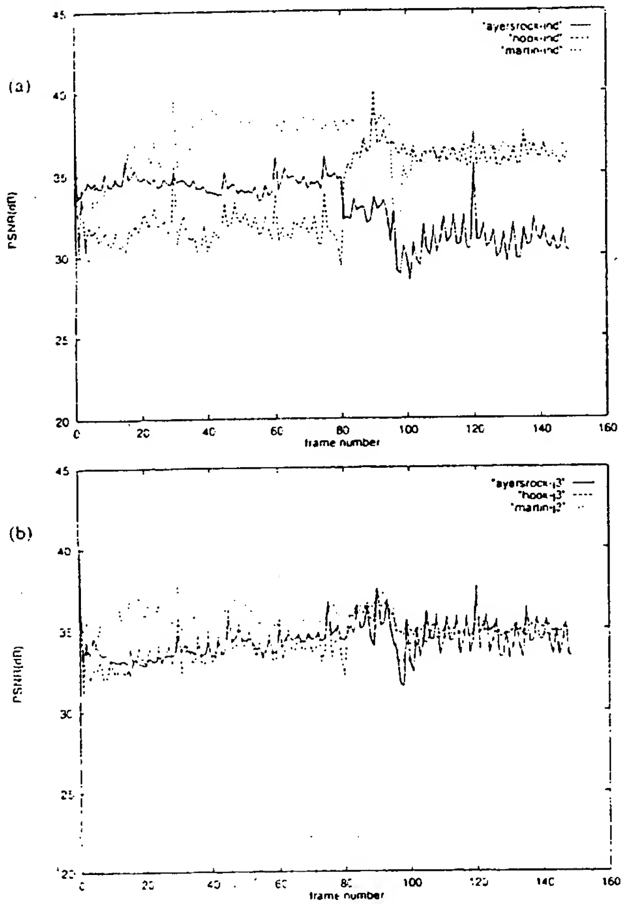

Fig. 11. PSNR (dB) for three sequences (*Ayers, Hook,* and *Martin*). (a) Independent coding, each at 3 Mbits/s. (b) Joint coding at a total rate of 9 Mbits/s.

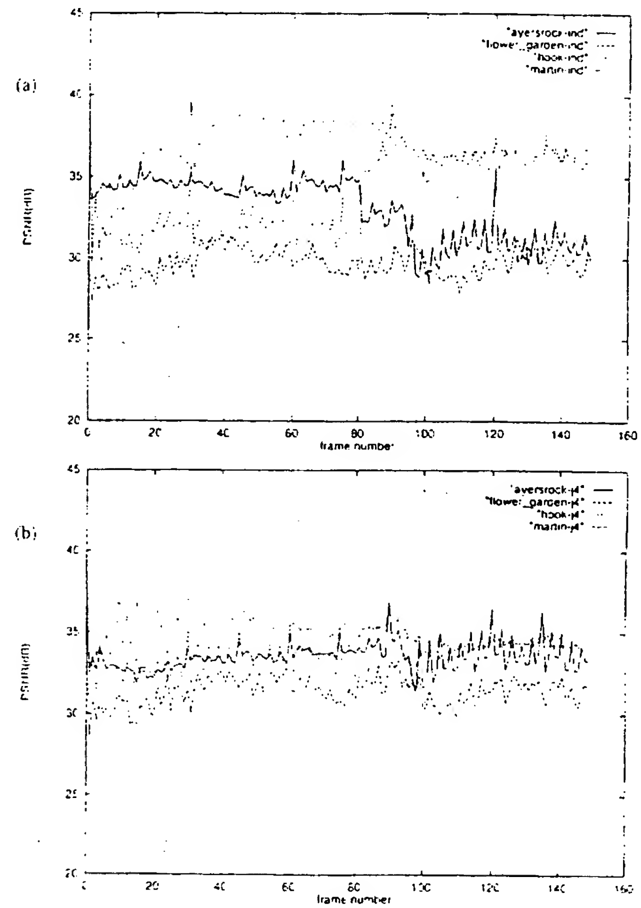

Fig. 12. PSNR (dB) for four sequences (*Ayers, Flower, Hook,* and *Martin*). (a) Independent coding, each at 3 Mbits/s. (b) Joint coding at a total rate of 12 Mbits/s.

coding than with independent coding. Also notice the smaller variability in PSNR within each program by joint coding as compared to independent coding.

IV. CONCLUSIONS

In this paper, we examined the bit-allocation issue for joint coding of multiple video programs and formulated a bit-allocation strategy of a hierarchical nature. At the very top level, we introduced the concept of a super GOP so that the programs can be reorganized into a set of identical groups of super GOP's in terms of the number of frames of each picture type. The identical super GOP's are assigned the same number of bits. We further defined a super frame as a collection of frames, each from one of the programs taken at the same time instant. We formulated the bit-allocation strategy for the super frames within a super GOP, given a target number of bits for the super GOP. We then formulated the bit-allocation strategy for regular frames in each super frame for a given target bit rate for the super frame. To prevent the encoder and decoder buffers from overflowing and underflowing and to limit each individual bit rate within a specific range, we added

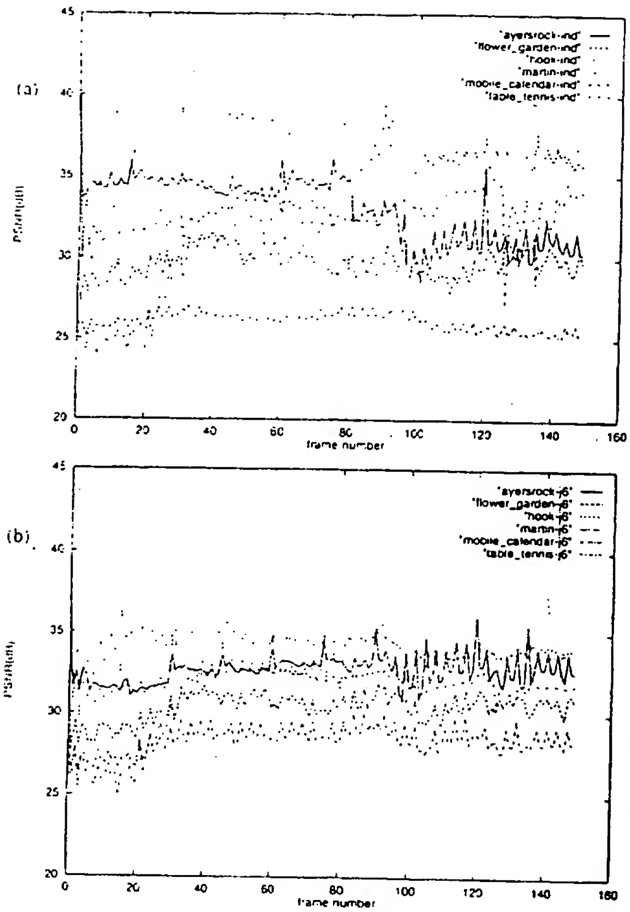

Fig. 13. PSNR (dB) for six sequences (*Ayers, Flower, Hook, Martin, Mobile,* and *Tennis*). (a) Independent coding, each at 3 Mbits/s. (b) Joint coding at a total rate of 18 Mbits/s.

constraints on the target bit rates for both super frame and regular frame. To meet the target bit rate for each frame, we used the rate-control scheme described in [9]. The simulation results demonstrate that joint coding, using the proposed bit-allocation strategy, results in more uniform picture quality among programs as well as within a program, compared to independent coding. In other words, joint coding is able to improve channel utilization by dynamically distributing the channel capacity among programs according to their respective complexities. Furthermore, the benefits of joint coding are achieved even with a small number of programs.

ACKNOWLEDGMENT

The authors wish to thank P. Blanchfield of the Communications Research Centre for his valuable help in conducting the computer simulations.

REFERENCES

[1] "Generic coding of moving pictures and associated audio," ISO/IEC (MPEG-2), Mar. 1994.
[2] "Test model 5," ISO/IEC (MPEG-2), Apr. 1993.
[3] T. Koga, Y. Iijima, K. Iinuma, and T. Ishiguro, "Statistical performance analysis of an interframe encoder for broadcast television signals," *IEEE Trans. Commun.*, vol. C-29, pp. 1868–1875, Dec. 1981.
[4] A. Guha and D. J. Reininger, "Multichannel joint rate control of VBR MPEG encoded video for DBS applications," *IEEE Trans. Consumer Electron.*, vol. 40, pp. 616–623, Aug. 1994.
[5] G. Keesman and D. Elias, "Analysis of joint bit rate control in multi-program image coding," in *Proc. SPIE VCIP*, Sept. 1994, pp. 1906–1917.
[6] L. Wang and A. Vincent, "Joint rate control for multi-program video coding," *IEEE Trans. Consumer Electron.*, vol. 42, pp. 300–305, Aug. 1996.
[7] ——, "Joint coding for multi-programs transmission," *Proc. IEEE Int. Conf. Image Processing*, Sept. 1996, pp. 425–428.
[8] A. R. Reibman and B. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 361–372, Dec. 1992.
[9] L. Wang, "Rate control for MPEG video coding," *Proc. SPIE Visual Communications and Image Processing*, Taiwan, May 1995, pp. 53–64.
[10] W. Verbiest and L. Pinnoo, "A variable bit rate video codec for asynchronous transfer mode networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 761–770, June 1989.

Limin Wang received the Ph.D. degree in electrical engineering from University of Ottawa, Ottawa, Ont., Canada, in 1988.

He is currently a Senior Staff Engineer in the Department of Advanced Technology, General Instrument Corp., San Diego, CA. Previously, he was with the Medical Communications Research Centre (1988–1990), MPR Teltech Ltd. (1990–1991), and the Communications Research Centre (1991–1997). His research interest is in the general area of image and video processing and transmission. He has published more than 50 papers in journals and conferences.

André Vincent received the B.Sc. degree in electrical engineering from l'École Polytechnique de Montréal, Montréal, P.Q., Canada, in 1975.

From 1975 to 1977, he was with the Department of National Defence, where he worked on the design of maritime communications systems. From 1977 to 1979, he was with Canadian Marconi, where he was involved in the design and development of mobile radio communications systems. He joined the Communications Research Centre, Ottawa, Ont., Canada, in 1979, where he conducted research in the areas of Teletext, data transmission, television channel characterization, and digital mobile radio. Since 1986, he has been involved in research in video processing, video compression, HDTV, and advanced television systems. He is currently Manager of Advanced Video Systems at the Communications Research Centre.